J. Kempe,[1,2], D. Bacon,[1,3] D. P. DiVincenzo,[4] and K. B. Whaley[1]
Departments of Chemistry[1], Mathematics[2] and Physics[3], University of California, Berkeley 94270
[4]*IBM Research Division, T.J. Watson Research Center, Yorktown Heights, New York 10598*

We present a theoretical analysis of the paradigm of encoded universality, using a Lie algebraic analysis to derive specific conditions under which physical interactions can provide universality. We discuss the significance of the tensor product structure in the quantum circuit model and use this to define the conjoining of encoded qudits. The construction of encoded gates between conjoined qudits is discussed in detail. We illustrate the general procedures with several examples from exchange-only quantum computation. In particular, we extend our earlier results showing universality with the isotropic exchange interaction to the derivation of encoded universality with the anisotropic exchange interaction, i.e., to the XY model. In this case the minimal encoding for universality is into qutrits rather than into qubits as was the case for isotropic (Heisenberg) exchange. We also address issues of fault-tolerance, leakage and correction of encoded qudits.

I. INTRODUCTION

Many machines have been invented to perform certain specific computational tasks. However in the language of computer science, these machines are not necessarily *computers*, meaning that they can not perform *all possible* computational tasks, i.e., they are not *universal*. What is considered as *all possible computations* depends on the underlying laws that govern the machines we use. In the classical world we can cast any computational task into a *boolean function* of the input (in bits). The evaluation of a *computable* function can then be reduced to a sequence of elementary logical operations acting on a few bits each: *Classical circuits* can be build from a *universal gate set*. The computational task of a quantum computer is to accurately perform *any unitary* operation on the input (qubits). These unitaries will have to be constructed from the basic units the computer will be equipped with, i.e., from a *universal quantum gate set*. Moreover these realistic quantum gates have to be physically realizable - they should involve only a few qubits at a time, for example.

Research efforts in recent years has shown that realistic *universal quantum gate sets* exist and that uniform *quantum circuits* can be built. A physical realization needs to implement such a universal set in order to be considered a universal quantum computer. Some of the gates in such a set are easier to implement than others. In several proposals some two-qubit gates are much easier to implement than one-qubit gates, yet the latter ones are needed to complete a universal set of gates. We have previously presented results on how to make do with a restricted set of gates that is not universal. This restricted set does not accurately enact every unitary operation on the system qubits, yet can nevertheless provide universality on a restricted subspace [1,2]. This approach, that we have termed "encoded universality", employs the philosophy of *encoding* the information, in order to make an a priori non-universal set of gates universal on the code [3]. This potentially reduces the experimental efforts of implementing gates that are hard to build.

In this paper we show that our recent demonstration of universality from a single physical interaction achieved with the Heisenberg exchange interaction [2] can be extended to an anisotropic interaction, namely to the XY model. This well-known Hamiltonian model is relevant to a number of two-dimensional condensed matter phenomena. The result shown here indicates the generality of the approach of seeking universal encodings tailored to specific physical interactions [3].

A second aim of this paper is to expand the Lie algebraic analysis presented briefly in Ref. [3], to show specifically the conditions under which selected elements of the algebra generated by the physical interactions can provide universality. We distinguish between the general results of universality on a d-dimensional subspace provided by this approach [2], and a more practical attempt to develop encoded one- and two-qubit gates that implicitly imposes the standard model of arbitrary one-qubit rotations and CNOT gates on the encoded qubits [1,2]. The construction of these encoded gates is discussed in detail, including efficient implementation in terms of discrete operations [4]. We will also address the important issue of fault-tolerance and leakage detection and correction. This will furnish a complete picture of how *fault-tolerance* meshes with encoded universality.

The structure of this paper is as follows. We shall first provide a brief summary of the concept of universality within the quantum circuit model of computation in Section II. We then present the essentials of the Lie algebra analysis underlying the problem of simulation of a quantum circuit, and discuss the various levels of tensor product structure that are required to implement a quantum circuit structure. The representation theoretic framework is

presented in Section III, illustrated by a summary of the earlier result for isotropic exchange [2,4,3]. We also provide the mathematical connection to results for universal computation on decoherence-free subsystems [2]. The core of the paper follows in Section IV, with a detailed presentation of the derivation of universal encodings for the anisotropic exchange interaction. Finally we address the issues of fault-tolerance and leakage (Sec. V) and close with discussion of some open topics (Sec. VI).

II. UNIVERSALITY

A. Quantum Circuits and Complexity

To build a computer from a physical system for algorithmic applications means to provide its user with a set of "units" from which he or she can implement an algorithm. A classical computer has to evaluate boolean functions. A convenient model for classical computation is the *uniform* circuit-model. The input to the circuit is given by a string of bits; wires carry them to elementary gates (such as AND and NOT) which are again interconnected by wires. The computation carries the bits along the wires and executes the gates, when they are reached. The output wires carry the results. These circuits can be parametrized by the size of the problem, i.e. the number n of input bits. Informally speaking such a family of circuits is *uniform* if given n, the corresponding circuit can be constructed efficiently (i.e. in polynomial time in n). This assures, that the complexity of a problem is not hidden into the design of the circuit. Such uniform circuits can be simulated at polynomial cost by a *universal circuit*, one of which is the circuit with the $NAND$ and the $COPY$ gates only [5]. The notion of complexity defines which problems are hard and which are easy according to the amount of space and time (i.e. gates) a circuit uses to solve them. Although classical universal circuits may differ in the basic gate-sets they use, any universal circuit can simulate any other *efficiently*, i.e. with only polynomial overhead in both space and time. Consequently the definition of complexity of a circuit can be made independently of the specifics of the gate-set used.

The Quantum Circuit Model (QCM) can be viewed as a transposition of the classical circuit model to the quantum world. A quantum computer implements unitary operations. The input is given by a string of *qubits*, connected to elementary gates. These gates constitute a universal set if the circuit can implement any unitary operation (and simulate any other circuit). A measurement on the output wires gives the result of the computation. Note that the QCM comes equipped with a canonical decomposition into a tensor product of small systems - the qubits (or qu-*d*its in the general case of d dimensions). This is of course not only necessary to define a basic unit (like a bit for a classical computer), but also to decompose a circuit into basic gates. Both are indispensable for the uniformity of the circuit family: given a size of a problem (e.g. to factor a number given as a bitstring of length n) one should be able to efficiently (i.e. polynomial in n) find the necessary gate sequence. This would be impossible if the structure of the Hilbert space changed with increasing n. The tensor structure ensures that this is not the case, and the QCM is thereby amenable to analysis in terms of the *quantum complexity* of an algorithm [6].

We note here, as pointed out by many other authors [7–9], that all known physical models which can perform some form of quantum computation appear to be simulatable by the quantum circuit model. We therefore take this empirically verified hypothesis as a fundamental definition of what it means to build a quantum computer. Thus, for the issue of universality, some map between our physical system and the QCM must be established. In particular, one of the essential features of the QCM is a tensor product structure between different subsystems and therefore we shall impose this on our physical system.

B. Universal Gate Sets

In both the classical and quantum regime, *universality results* have been established that ensure the equivalence of ostensibly different models of computation which may employ different sets of gates. Studies of universality in the QCM have culminated in the definition of a *universal gate set*. A natural impulse is to make a close analogy to the classical case, i.e. to search for a finite set of gates that *exactly* implements every quantum circuit efficiently. However, these attempts are doomed to failure: obviously a discrete set of gates can't be used to implement an arbitrary unitary operation *exactly*, since the set of unitary operations is continuous. Powerful results have nevertheless been established for the *approximate* simulation of quantum circuits with a set of finite gates, and we will now briefly review the steps leading there. A more detailed account can be found in [10].

To quantify the accuracy of an approximation, one needs a distance on matrices. On the finite N dimensional matrix space, any metric is basically as good as any other (i.e. they differ by only constant factors). For example,

we can use the trace-norm $d(\mathbf{U},\mathbf{V}) = \sqrt{1 - \frac{1}{N}\mathrm{Re}\left[\mathrm{Tr}(\mathbf{U}^\dagger\mathbf{V})\right]}$, or the norm $d(\mathbf{U},\mathbf{V}) = max_{|\Psi\rangle}\|(\mathbf{U}-\mathbf{V})|\Psi\rangle\|$ ($|\Psi\rangle$ is a normalized state). A matrix $\mathbf{V}$ is then said to approximate a transformation $\mathbf{U}$ to accuracy ϵ if $d(\mathbf{U},\mathbf{V}) \leq \epsilon$.

The methodology employed in most universality proofs and constructions starts with a set of *gates* and then operates within the *unitary group* generated by their repeated applications to show that the whole of $SU(N)$ is generated, where N is the dimension of the space[1]. Historically the first universal quantum gate sets were three-qubit gates (Deutsch [8]). Earlier work by Toffoli [12] on classical circuits has introduced a classical universal three-bit gate - the *Toffoli-gate* - which is reversible. Classical circuits based on the Toffoli-gate can simulate all other classical circuits efficiently and in a reversible way. Deutsch generalized this gate to a unitary three-qubit quantum gate and proved its universality. In 1994-96 several groups improved on Deutsch's circuits. DiVincenzo [11] demonstrated how to construct Deutsch's three qubit gate using only one and two qubit gates, and thus demonstrated that the physically unrealistic three qubit interactions of Deutsch are not necessary for universality. Barenco *et al.* [13] proved that single qubit gates together with the controlled not (CNOT) allow one to *exactly* implement any unitary transformation. This is now so often taken as the paradigm for universal computation that we may refer to it as the "standard model". Another memorable universality result is due independently to Deutsch, Barenco and Ekert [14] and Lloyd [15] in 1995. They prove using Lie-algebra arguments (see next section) that *almost any* two-qubit unitary gate is universal. *Almost any* is to be taken in the generic sense: in the 16-dimensional manifold of possible two-qubit unitary gates acting between n qubits, the set of gates which *do not* yield a universal set of gates is a lower dimensional manifold. They implicitly assume that the generic gates (given by Hamiltonians H_{ij}) can be applied between all qubit pairs i,j and in both directions (i.e. as H_{ij} and H_{ji}), i.e. they assume that some sort of permutation (or exchange) gate is given. However many of the accessible interactions in physical systems are not of this sort - nature is often *non-generic*!

To quantum compute in the presence of (inevitable) errors, quantum information has to be protected against noise. To this end several approaches have been developed, among which are the theory of quantum error correcting codes (QECC) [16] and the theory of decoherence-free subspaces and systems (DFS) [17,18]. In these approaches quantum information is encoded into parts of the system Hilbert space. The theory of *fault-tolerant* quantum computation shows how to achieve universality in the presence of errors and imperfect quantum gates. The basic idea is to work directly with encoded universal gates that act on an encoded space, whether this derives from a quantum error correcting code or from a decoherence-free subspace or subsystem. Considerations of fault-tolerance have led to a number of specific universal gate sets. These gate sets are useful because it is known how to implement them within the context of fault-tolerant quantum error correction. We will mention only one of them here [19], namely the set consisting of $\{H, P, CNOT, \pi/8\}$ where

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad \frac{\pi}{8} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \quad CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1}$$

For practical implementations, the degree of locality of the gates is important. For QECCs it has been shown how to implement such an encoded discrete universal set of gates using local gates on the physical qubits [16]. These gates take encoded states to encoded states, but may take the state outside of the encoded subspace during the computation. In order to apply this approach, the speed of gate operations is critical: it has to be less than the error rate. In the theory of computation on DFSs, Bacon et al. [1] (for a specific class of DFSs) and Kempe et al. [2] (on a broader class of DFSs) have shown how compute on DFS with local Hamiltonians only. This work shows that the two-body exchange Hamiltonian alone is sufficient to implement universal computation on a decoherence-free subsystem. This demonstrates in particular, that there is an *encoding* of the quantum information into a subspace such that the exchange Hamiltonian - which is a priori not universal in the sense that it can perform any unitary evolution on all qubits - does now *become* universal. We have termed this "encoded universality". Ideas about encoded universality in the context of physical realizations of a quantum computer have been presented in [3,4,20,21] and also implicitly in [22] in the context of dynamical noise suppression schemes (so called "bang-bang" control).

---

[1] A notable exception is the proof by DiVincenzo [11]

C. Lie Analysis of Intrinsic Physical Interactions

Every quantum gate is generated by a Hamiltonian. It is very convenient to recast the universality results into a Hamiltonian language. This allows us to use Lie-algebra analysis to approach universality constructions. Physically, the implementation of a unitary quantum gate will come about due to the implementation of some controllable Hamiltonian for a specific time.

Instead of supposing that we are given a set of unitary gates, we thus rather assume that we have control over a set of Hamiltonians which we can apply to our system for a controllable amount of time. Of course, there will be inaccuracies in the application of a Hamiltonian for any given exact time. These inaccuracies, however, are exactly what fault-tolerant quantum computation is designed for. Consequently it is legitimate to proceed from the assumption that we can exactly implement a given Hamiltonian for a given amount of time. The aim is then to a) reformulate the analogue of universal sets of unitary gates Hamiltonians formulation and b) to provide efficient finite time approximations that result in implementable sets of universal gate operations.

We can now ask what new gates (and new Hamiltonians) we can obtain by operating these given ones. A general theoretical procedure to combine known gates to construct new gates is well known, deriving from the Baker-Hausdorff-Campbell operator expansion. Accordingly, we can combine two Hamiltonians to construct new Hamiltonians using the following properties:

$$e^{i(\alpha\mathbf{A}+\beta\mathbf{B})} = \lim_{p\to\infty}(e^{i\alpha\mathbf{A}/\mathbf{p}}e^{i\beta\mathbf{B}/\mathbf{p}})^p \tag{2}$$

$$e^{i[\mathbf{A},\mathbf{B}]} = \lim_{p\to\infty}(e^{-i\mathbf{A}\sqrt{\mathbf{p}}}e^{i\mathbf{B}/\sqrt{\mathbf{p}}}e^{i\mathbf{A}/\sqrt{\mathbf{p}}}e^{-i\mathbf{B}\sqrt{\mathbf{p}}})^p. \tag{3}$$

In practice we will truncate the infinite series to approximate the relevant operations. A detailed analysis of this is provided in [15,2]. As a consequence of these composition laws scalar multiples, sums and Lie-commutators $i[\mathbf{A},\mathbf{B}]$ can be obtained out of the given Hamiltonians). Note that these operations correspond to closing the set of allowed Hamiltonians as a *Lie-algebra*. To approximate to a given accuracy, say $\exp(i(\alpha\mathbf{A}+\beta\mathbf{B}))$, by a sequence of gates with generators $\mathbf{A}$ and $\mathbf{B}$ one just truncates Eq. (2) for large enough p. This allows us to approximate every unitary element in the *Lie-group* corresponding to the Lie-algebra generated by the given Hamiltonians using controllable coupling constants. The inverse of this statement is also true, namely all gates that we can build of a set of given interactions are generated by elements in the Lie-algebra.

It is now easy to define a *universal set of generators* H in this setting: H is universal if the Lie-algebra generated by the elements of H contains $su(N)$, where N is the dimension of the underlying space and $su(N)$ is the Lie-algebra of the special unitary group. This implies that the set of *unitary* gates obtainable via successive application of *gates* deriving from H is dense in the group $SU(N)$.

This general approach provides a systematic way to construct universal sets of gates, but so far it says nothing about whether or not the resulting gates can be implemented efficiently. We remind the reader that we are using the term "efficiently" in the quantitative sense defined above, namely in terms of whether implementation can be made with only polynomial overhead in the number of discrete gates required. The Trotter expansion, Eq. (2), is guaranteed to be accurate in the limit as the number of terms p becomes very large, but clearly a critical practical issue is how the truncation scales with the size of the quantum circuit, n. Thus in order to actually use this approach to universality for realistic physical implementation, it is crucial to have bounds on the length of the gate sequences approximating a certain gate in terms of the desired accuracy for a circuit of given size. This is all the more important if one universal set is to be replaced by any other with only polynomial overhead in the number of gates applied, for otherwise the complexity classes would not be robust under the exchange of one set for another. The Solovay-Kitaev theorem[2] establishes the equivalence of universal sets, and provides bounds on the length of gate sequences for a desired accuracy of approximation. In short, this theorem states:

> *Theorem (Solovay-Kitaev):* — Given a set of gates that is dense in $SU(2^k)$ and closed under Hermitian conjugation, any gate $\mathbf{U}$ in $SU(2^k)$ can be approximated to an accuracy ϵ with a sequence of poly $[\log(1/\epsilon)]$ gates from the set.

It follows that any quantum circuit (performing any desired unitary operation) can be simulated to arbitrary accuracy efficiently by a circuit with gates from a universal set. In the context of a universal set of Hamiltonians H it implies that all the gates in the Lie-group can be obtained efficiently by an appropriate choice of the control parameter. This powerful result lays open the route to searching for radically different universal sets of operators.

---

[2]For details and proofs see [10]

III. FORMALISM

Designs for quantum computers can come from a variety of underlying physical systems. A major challenge for future quantum engineers is to find the physical systems that allow for a universal set of gates. In all the physical implementations proposed only particular Hamiltonians can be turned on and off. As mentioned in Sec. II B, generic two-body interactions can suffice to implement any unitary transformation. Nature, however, is not generic. Most interactions in real physical system possess symmetries that place these interactions squarely on a lower dimensional "non-generic" manifold where they are not universal. Thus, while allowing in principle for universality based entirely on two-body interactions, this approach does not provide a general prescription for universality given a specific set of intrinsic interactions.

We propose a different approach here, namely to start from the natural interactions given by the physics of the proposed qubit system, and then to investigate the underlying potential of these interactions for "encoded universality". The rationale behind this is that the given interactions - although *per se* not universal - become *universal within a subspace*. Encoding into this subspace allows to reduce the number of different Hamiltonians needed for universality, at the expense of spatial resources. This approach allows one to take the natural physical constraints of a specific implementation into account. For a given physical system, the natural interactions of choice can then be determined by various factors, including their speed, the ease with which they can be implemented, any engineering or material constraints, and their robustness towards decoherence processes.

Here we detail this new paradigm of "encoded universality". We start by reviewing the relevant results from our previous work addressing quantum computation on decoherence-free subsystems and by summarizing the mathematical framework. We illustrate the general theory with a summary of our previous results on the isotropic exchange interaction [2,4,3], before presenting the new results for the anisotropic exchange interaction in Section IV.

A. Representation Theoretic Framework

The general formalism employed to find universal encodings is largely related to the Lie-algebra framework for *noiseless* or *decoherence-free* (DF-) subsystems [23]. We briefly review this framework here.

In the context of DF- subsystems [23] we are dealing with operators in the interaction Hamiltonian $\mathbf{H}_I = \sum_\alpha \mathbf{S}_\alpha \otimes \mathbf{B}_\alpha$. The Hamiltonian $\mathbf{H}_I$ describes the system-bath interaction which ultimately causes decoherence and corruption of the information stored in the system. The S_α (acting on the system) generate an *interaction algebra* $\mathcal{S}$, which is the algebra generated by the set $\mathcal{S}_1 = \{I, S_1, S_2, \ldots\}$ by *linear combination* and operator *multiplication*. Because the interaction Hamiltonian is Hermitian, this algebra is †-closed. The algebra $\mathcal{S}$ splits into irreducible components as a consequence of the following general theorem from the representation theory of †-closed algebras (see, e.g., [24]):

Theorem 1 *Let $\mathcal{S}$ be a †-closed algebra of operators acting on a space $\mathcal{H}_S$. Then the space is isomorphic to a direct sum*

$$\mathcal{H}_S \sim \sum_{J\in\mathcal{J}} \mathbb{C}^{n_J} \otimes \mathbb{C}^{d_J} \tag{4}$$

in such a way that in this representation $\mathcal{S}$ and the commutant $\mathcal{S}'$ of $\mathcal{S}$ are decomposable respectively, as

$$\mathcal{S} \cong \bigoplus_{J\in\mathcal{J}} I_{n_J} \otimes M(\mathbb{C}^{d_J}) \qquad \mathcal{S}' \cong \bigoplus_{J\in\mathcal{J}} M(\mathbb{C}^{n_J}) \otimes I_{d_J} \tag{5}$$

.

Here $M(\mathbb{C}^n)$ means the set of *all* linear operators from $\mathbb{C}^n$ to itself, and $J \in \mathcal{J}$ runs over all the irreducible representations (of dimension d_J and with degeneracy n_J) of $\mathcal{S}$. The commutant $\mathcal{S}'$ of $\mathcal{S}$ is the space of all operators commuting with $\mathcal{S}$. It is itself an *associative algebra*, i.e. an algebra closed under linear combination and multiplication. States encoded in $\mathbb{C}^{n_J}$ are completely immune to the interaction with the bath, because the interaction algebra $\mathcal{S}$ acts only on its "co-factor" $\mathbb{C}^{d_J}$. Thus $\mathbb{C}^{n_J}$ is a *noiseless* or *decoherence-free* subsystem where information is intrinsically stabilized against the effects of the interaction with the bath, with no need for corrective action. It is important to note that the *Lie-algebra* generated by the $\mathbf{S}_\alpha$ via *linear combination* and *Lie-commutator* is different from the interaction algebra $\mathcal{S}$. This point will become important later.

One of the most important physical models of decoherence, which allows for decoherence-free subspaces and subsystems, is the model of so called *collective decoherence*. In the case of collective decoherence on n qubits the *interaction*

algebra $\mathcal{S}$ is spanned by $\mathbf{S}_\alpha = \sum_{i=1}^n \sigma_\alpha^i$, with $\alpha = x, y, z$. This algebra is well known in the quantum mechanical literature, it represents a (reducible) representation of the Lie-algebra of $su(2)$. Its block-structure according to Theorem 1 allows to find the decoherence-free encoding of the information.

Yet this encoding alone leads only half the way towards decoherence-free quantum computation. In order to harness DF-subsystems as quantum computers it has to be shown that one can perform universal computation *on DFSs* in a fault-tolerant way using local gates on the physical qubits. Inspecting the statement of Theorem 1 more closely it is not hard to see that operations that take DF-states to DF-states must be generated by Hamiltonians which lie in the commutant $\mathcal{S}'$ of the interaction algebra $\mathcal{S}$. For universal computation within a DF-subsystem we need to identify local gates in this commutant (these will be the Hamiltonians we can control) and to show that the Lie-algebra generated by these Hamiltonians contains $su(n_J)$ on each cofactor $\mathbb{C}^{n_J}$.

In our work on fault-tolerant universal computation on decoherence-free subsystems we showed [1,2] that the exchange interaction alone generates the Lie-algebra su on every such DF-subsystem. This result was obtained for any set of n qubits, $n \geq 3$ and settled the question of universal fault-tolerant computation on decoherence free subsystems with local interactions.

Let us now switch viewpoints from consideration of the subsystem to consideration of the interaction. Our result [1,2] then implies that if we restrict the Hilbert space of the system to the space of the subsystem only, the exchange interaction *alone* is universal. In other words, the encoding into subsystems provided a way to make the exchange universal, meaning that the set of exchange interactions $\mathsf{E} = \{\mathbf{E_{ij}} : 1 \leq i < j \leq n\}$ on n qubits is a *universal set of generators over a subspace* according to the definition in Sec. II C! We point out that while this result was achieved specifically for the decoherence-free subsystems that are defined by resistance to collective decoherence, the decoherence-free properties of the subsystems are not required for the universality result.

The scenario of *encoded universality* is in a sense the "inverse" problem to universal computation on DFSs. In contrast to the DFS situation, we are now presented with a set of given interactions H on the system Hilbert space $\mathcal{H}_S$. The operations we can implement from this set generate a *Lie-algebra* $\mathcal{L}(\mathsf{H})$. The *Lie-group* generated by this Lie algebra (via "exponentiation") is a subgroup of the unitary group U and thus is a compact Lie group. Compact Lie groups, along with the Lie algebra which generates the Lie group, are completely reducible[3] and hence the Lie-algebra splits into a direct sum of irreducible representations (irreps)

$$\mathcal{L}(\mathsf{H}) \cong \bigoplus_{J \in \mathcal{J}} \mathcal{L}_J(n_J) \otimes I_{d_J} \tag{6}$$

over the state space

$$\mathcal{H}_S \sim \sum_{J \in \mathcal{J}} \mathbb{C}^{n_J} \otimes \mathbb{C}^{d_J}. \tag{7}$$

Here $\mathcal{L}_J$ is the Jth irrep of $\mathcal{L}(\mathsf{H})$. This decomposition is reminiscent of that of the commutant $\mathcal{S}'$ in Eq. (5), and so we use the same variables for the dimension n_J of the Jth irrep of $\mathcal{L}_J$ and its degeneracy d_J. (Note that these variables are interchanged for the interaction algebra $\mathcal{S}$.)

The problem we face in order to make the set H universal can now be formalized as follows:

Given the Lie algebra $\mathcal{L}(\mathsf{H}) \cong \bigoplus_{J \in \mathcal{J}} \mathcal{L}_J(n_J) \otimes I_{d_J}$ *find a component* $\mathbb{C}^{n_J} \otimes \mathbb{C}^{d_J}$ *of the state space over which* $\mathcal{L}_J(n_J)$ *contains* $su(n_J)$.

If this can be satisfied, then we have achieved universality on a subsystem.

The exchange-only universality theorem we proved in [2] shows that the *Lie-algebra* generated by the exchange interaction generates $su(n_J)$ on *each* component $\mathbb{C}^{n_J}$. This implies that exchange lends itself for encoded universality. We can encode into any of these components (corresponding to subsystems in the DFS framework). The problem of the existence and structure of "encoded universality" with isotropic exchange is thus solved in the affirmative.

Furthermore, we can immediately infer the coding efficiency of the "encoded universality" obtained from exchange. As shown, e.g., in [2] the number of encoded qubits k over the number of physical qubits n approaches 1 as $n \to \infty$. This means that for larger and larger n the amount of redundancy in physical qubits that are needed to encode approaches zero. However, as was outlined in Sec. II A, we do have to introduce a tensor product structure via what we will call *conjoining* (see Sec. III C), and this will place a constraint on the encoding efficiency.

---

[3] A representation of a Lie algebra is irreducible if the action of the representation on its vector space does not possess an invariant subspace.

The tensor product structure is important in order to map the QCM onto our subspace . The convenient way to proceed in construction of this structure is to use "clusters" of subspaces over which exchange is universal. These clusters may be delimited either by some natural length scale in the physical system, or by an arbitrary cut-off. For instance the *smallest* possible block size consists of 3 physical qubits (to encode two states - a logical qubit) (see Sec. III D). The exchange interaction allows us to perform operations both within and between the blocks. The encoding efficiency with such a tensor product structure is then limited by the block size to a value less than unity, no matter how many qubits are employed. Choosing a higher cut-off, *i.e.*, a larger block size does increase the coding efficiency, but this will always remain at a value less than unity for finite block size. We will give more details of the conjoining procedure in Sec. III C.

We will briefly review some specific examples for isotropic exchange in Sec. III D. In Sec. IV we will show that a similar result also holds for the XY-interaction, *i.e.*, for an anisotropic exchange interaction.

B. Interaction Algebra ≠ Lie-Algebra

The algebra representation Theorem 1 has given rise to some misunderstanding, which we now clarify. Let us look at the case of collective decoherence on n qubits. The *interaction algebra* $\mathcal{S}$ is spanned by $\mathbf{S}_\alpha = \sum_{i=1}^n \sigma_\alpha^i$, with $\alpha = x, y, z$, via linear combination and *multiplication*. The *Lie-algebra* spanned by the $\mathbf{S}_\alpha$ - we will denote it by $\mathcal{L}(S)$ - represents a (reducible) representation of the Lie-algebra $su(2)$. This is *different* from $\mathcal{S}$, since it is obtained by linear combination and *taking commutators*, i.e. in general $\mathcal{L}(S) \subseteq \mathcal{S}$. However, for compact groups (and the S_α form a compact group), it is true that when expressed in the same basis as $\mathcal{S}$, the *Lie-algebra* $\mathcal{L}(S)$ splits into the same block-structure as that of Eq. (6).

As we can easily infer from the DFS theorem [25,2], if the encoded operations are to preserve the information in the encoded subsystems, the gate Hamiltonians should be in the commutant $\mathcal{S}'$ of $\mathcal{S}$. It is exactly these operations in $\mathcal{S}'$ that act entirely within the subsystems, as can be seen from Eq. (5). Yet it turns out that the commutant of $\mathcal{L}(S)$ and the commutant of $\mathcal{S}$ are the same; in both cases it consists of the set of elements that commute with all the $\mathbf{S}_\alpha$.

The good news is that instead of all $\mathcal{S}$, we only need to study the structure of the irreps of $su(2)$ as a *Lie-algebra* $\mathcal{L}(S)$. It is very well known in quantum-mechanics since it is precisely the Lie algebra deriving from angular momentum. This will give us the noiseless or decoherence-free subsystems, and also the commutant, $\mathcal{S}'$. Note that the Lie algebra $su(2)$ arises specifically from the DFSs resulting from the collective decoherence model. Other decoherence models will yield different DFSs that may be defined by different Lie algebras.

As is well known from the representation theory of $su(2)$ (see e.g. [26]) the commutant of $\mathcal{L}(S)$ is related to the *symmetric group* (permutation group) S_n [25]. The *natural representation* of S_n on n qubits is the set of operators that permutes the qubits, i.e. if $\pi \in S_n$, then π acts on basis states as

$$\pi : |i_1\rangle|i_2\rangle \dots |i_n\rangle \longrightarrow |i_{\pi(1)}\rangle|i_{\pi(2)}\rangle \dots |i_{\pi(n)}\rangle \qquad i_k \in \{0,1\} \tag{8}$$

Clearly the elements of S_n commute with the permutation-invariant $\mathbf{S}_\alpha$. But they - together with the associative algebra they generate - also constitute the set of *all* elements that commute with them. In other words the commutant of $\mathcal{L}(S)$ (this is equal to $\mathcal{S}'$, the commutant of $\mathcal{S}$) is given by the algebra spanned by linear combinations of elements of S_n in its natural representation, Eq. (8). The *associative algebra* spanned by the permutation group S_n gives the set of *all* operators on a subsystem, as can be seen from Eq. (5). In particular, we can find a set of generators for *universal computation* on subsystems among its elements. This means that for universal fault-tolerant computation all we have to do is to implement the permutation-group. It is also known that the permutation-group S_n is generated by transpositions τ_{ij} (which just permute two qubits). These transpositions can obviously be implemented by local interactions - just switch on the exchange Hamiltonian. One might be tempted to conclude that the question of universal computation with local gates is thus settled!

Unfortunately this conclusion is false. The transpositions generate the permutation group S_n via *multiplication*. But the composition laws for given operations do not generate *products* of allowed Hamiltonians. The allowed ways to compose operations from a set of basic Hamiltonians give only *linear combinations* and *commutators*, i.e. they close the *Lie-algebra* of the basic set (Sec. II C). In other words, if our basic interactions are the exchange (transposition) of two qubits, we can implement the *Lie-algebra* generated by them, but not the associative algebra.

For universal computation with exchange we had therefore to show explicitly [2] that the Lie-algebra generated by the transpositions τ_{ij} generates the whole Lie-algebra of $su(n_J)$ on a subsystem $\mathbb{C}^{n_J}$.

In order to clarify this it is useful to discuss an example. Suppose that we are given a system of n qubits upon which we can enact one of the three collective Hamiltonians

$$\mathbf{C}_\alpha = \sum_{i=1}^n \sigma_\alpha^i. \tag{9}$$

The Lie algebra $\mathcal{A}$ generated by these interactions is simply that of $su(2)$:

$$[\mathbf{C}_\alpha, \mathbf{C}_\beta] = i\epsilon_{\alpha\beta\gamma}\mathbf{C}_\gamma. \tag{10}$$

This Lie algebra will thus be reducible to a set of n_J dimensional irreps of $su(2)$:

$$\mathcal{A} \cong \bigoplus_{J=0(1/2)}^{n/2} \mathcal{A}_J(n_J) \otimes I_{d_J}, \tag{11}$$

where $\mathcal{A}_J(n_J)$ is the $n_J = 2J+1$ dimensional irrep of $su(2)$ which appears with degeneracy d_J in the decomposition. It is important to realize that if a set of Hamiltonians represents an n_J dimensional irrep of $su(2)$, then the *operational* power of these interactions is not more than the operational power of any other irrep of $su(2)$. In particular, it is no more powerful than the two dimensional irrep of $su(2)$. An n_J dimensional representation of $su(2)$ acts on a n_J dimensional space, but does not enact $su(n_J)$ on this n_J dimensional space. Yet if we look at the *associative* algebra spanned by the $\mathbf{C}_\alpha$ and its decomposition into a block structure according to Theorem 1, we see that on each co-factor of each block this algebra generates every possible matrix, and in particular, all matrices in $su(n_J)$ (it is the same associative algebra that we studied in the case of collective decoherence). The point is that the power of the associative algebra is nevertheless not accessible via the fundamental physical composition laws.

C. Conjoining and the Tensor Product Nature of Computation

The above summary of representation theory for Lie algebras has been made solely in terms of some abstract d-dimensional Hilbert space. We now admit two tensor product structures: first, on our physical system, and second, on the encoded qubits from which encoded universality will be constructed.

The first tensor product structure is simply that implied by our physical system and is usually forced on us by the locality of physics. For example, if we are using the spins of single electrons on a quantum dot, the tensor product is just the natural one of these spin-qubits. The nature of this tensor product structure is also manifest in the set of interactions which will be present in the real world, and is represented in the Hamiltonians in H at our disposal. This tensor product is not relevant to our discussion except in the context that we expect the physical interactions we are dealing with to be *local* with respect to it.

The second tensor product is the important tensor product relevant to building a map between the given interactions and the QCM. As mentioned before *within* an *subspace for universality* there is a priori no tensor structure present. In order to build uniform quantum circuits for quantum computation we need to introduce a tensor structure at some point (Sec. II A). In particular we are seeking to use a certain set of interactions to simulate a QCM and therefore we require that there be a tensor product structure corresponding to the the tensor product structure in this model.

In order to define a tensor product structure, one must be able to identify subsystems with a tensor product structure between them. The exact method for establishing where this tensor product structure comes into play is arbitrary. From a physical perspective one is motivated to search for *small encodings*, i.e., we seek to minimize the number of physical qubits used to encode a logical qubit (or qu*dit* if d-dimensional). The tensor product structure will then be induced between these encoded qubits (qudits) by separating them into blocks (e.g., of nearest neighbors), with the defining property that single encoded-qudit operations are possible within each such block. However the cut-off is in principle largely arbitrary and so our ultimate choice will depend on the balance between a desired coding efficiency versus the complexity of gate sequences required for this. These blocks of encoded qubit (or qu*dit* if d-dimensional) systems will form the basic factors of our ultimate tensor structure. We term this process *"conjoining"* the encoded qudits [2]. Within each block we can implement any encoded operation as a consequence of our universality proofs.

A subtlety arises when we want to perform encoded operations *between* the blocks. This has to be addressed separately from the operations within a single block. In the case of the exchange interaction we have the property that the space formed by the two separate encoded qudits is itself part of an encoded higher dimensional system. This is a consequence of the structure of the subsystems, as shown explicitly in the proof in [2]. This means that the exchange interaction implements any operation within this bigger subsystem, including in particular, any operation we want to perform between the blocks. A similar statement is true in the case of the XY-interaction.

Finally, given a certain encoded universality construction, the issues of preparation and measurement on the encoded information must be established. Generally speaking, some method of efficiently generating states with a large overlap of the encoded information must be possible. Similarly, a method for extracting some bit of information from the encoded information must also be available. These are important issues which need to be addressed within the physical feasibility of a given encoded universality construction.

Finally we note that the formalism we have developed here works only for a quantum circuit where measurement is not used as a fundamental process for enacting unitary evolution. Recently it has been demonstrated that universal quantum computing can be achieved using only measurements on a particular prepared entangled state [27]. Our formalism does not deal with measurements used to construct approximate unitary evolution.

D. Example - Universality from Isotropic Exchange Hamiltonian

We will illustrate the theory of encoded universality with an example of our previous results on the isotropic exchange interaction [2]. The most general form of the Heisenberg exchange interaction between two spins is the fully anisotropic spin-spin coupling

$$\mathbf{H}_{ij} = J_{ij}^X \sigma_x^i \sigma_x^j + J_{ij}^Y \sigma_y^i \sigma_y^j + J_{ij}^Z \sigma_z^i \sigma_z^j. \tag{12}$$

In the isotropic limit the exchange couplings between a given ij pair are all equal, *i.e.*, $J_{ij}^X = J_{ij}^Y = J_{ij}^Z \equiv J_{ij}$, so that we are then dealing with the Heisenberg Hamiltonian

$$\mathbf{H}_{\rm Hei} = \sum_{i \neq j} J_{ij} \mathbf{E}_{ij}. \tag{13}$$

To show how this intrinsic coupling can lead to encoded universality, we analyze its action on three qubits, assuming for simplicity that the there is no dependence of the amplitude J on the qubit indices, i.e., $J_{ij} = J$.

The Lie algebra $\mathcal{L}_E$ generated by $\mathbf{E}_{ij}$ on three qubits allows us to implement the Hamiltonians in the set $\{\mathbf{E}_{12}, \mathbf{E}_{23}, \mathbf{E}_{13}, \mathbf{T} \equiv i([\mathbf{E}_{12}, \mathbf{E}_{23}]\}$. A better basis for the Lie algebra is given by the set of operators $\mathbf{H}_0 = \mathbf{E}_{12} + \mathbf{E}_{23} + \mathbf{E}_{13}$, $\mathbf{H}_1 = \frac{1}{4\sqrt{3}}(\mathbf{E}_{13} - \mathbf{E}_{23})$, $\mathbf{H}_3 = \frac{1}{12}(-2\mathbf{E}_{12} + \mathbf{E}_{23} + \mathbf{E}_{13})$, $\mathbf{H}_2 = i[\mathbf{H}_3, \mathbf{H}_1]$. We then find that

$$[\mathbf{H}_0, \mathbf{H}_\alpha] = 0 \tag{14}$$

for all α and

$$[\mathbf{H}_\alpha, \mathbf{H}_\beta] = i\epsilon_{\alpha\beta\gamma}\mathbf{H}_\gamma \tag{15}$$

with $\alpha, \beta, \gamma \in \{1, 2, 3\}$. $\mathbf{H}_0$ is an abelian invariant subalgebra of this Lie algebra and thus factors out as a global phase. The set $\{\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3\}$, on the other hand, act as the generators of $su(2)$. Thus the exchange interaction between three qubits can be used to implement a single encoded-qubit $su(2)$. More precisely, we find that the $\mathbf{H}_\alpha$ for $\alpha \in \{1, 2, 3\}$ generates the algebra

$$\mathcal{L}_E^{(3)} = (\mathcal{L}_1 \otimes I_4) \oplus (\mathcal{L}_2 \otimes I_2) \tag{16}$$

where $\mathcal{L}_d$ is the d-dimensional irrep of $su(2)$. Note the degeneracy of the corresponding irreps. Corresponding to this decomposition the $\mathbf{H}_\alpha$ act as

$$\mathbf{H}_\alpha = \mathbf{0}_4 \oplus \left(\frac{1}{2}\sigma_\alpha \otimes \mathbf{I}_2\right) \tag{17}$$

for $\alpha \in \{1, 2, 3\}$ where $\mathbf{0}_4$ is 4-dimensional zero operator (the 1 dimensional irreps all act as 0) and $\mathbf{I}_2$ is the two-dimensional identity operator. The action of the exchange is thus identical to that of an $su(2)$ operator on a single qubit when working over the encoded space defined by the above decomposition. If we encode our logical qubits as

$$\begin{aligned} |0_L\rangle &= \frac{1}{\sqrt{2}}(|010\rangle - |100\rangle) \\ |1_L\rangle &= \sqrt{\frac{2}{3}}|001\rangle - \sqrt{\frac{1}{6}}|010\rangle - \sqrt{\frac{1}{6}}|100\rangle \end{aligned} \tag{18}$$

then we find that the action of $\mathbf{H}_\alpha$ is $\frac{1}{2}\sigma_\alpha$. Due to the degeneracy of the irrep, other encodings are also possible. Note that these states are nothing but $J = 1/2$, $J_z = +1/2$ total angular momentum states of 3 spin-1/2 particles with a given projection along a certain axis, and can thus be found using elementary addition of angular momentum [2].

Having shown that the action of the exchange interaction on three qubits can produce the effect of a single 2-dimensional representation of $su(2)$, we now conjoin the qubits, i.e. induce a tensor product structure between blocks

of three qubits in order to simulate a quantum circuit. We thus *choose* an encoding scheme in which a single qubit is identified with 3 physical qubits. This is not the only choice of tensor product structure. In fact, any tensor product between sets of $k \geq 3$ qubits can be used to construct a universal gate set [2]. Once one has introduced a tensor structure for the encoding, it is necessary to show that the natural couplings of the system can produce a non-trivial action between the tensor components of this encoded tensor product structure, i.e., show that we get universal gates on the encoded qubits.

In the above case, where we have shown that we can obtain full control ($su(2)$) over an encoded qubit, what we now need to show is that a non-trivial action between the encoded qubits can be enacted. This is nothing more than a map from the encoded universality to the fully universal set of physical gates consisting of single qubit gates supplemented by a non-trivial coupling between the qubits (like the $CNOT$, for example). This point of the universality proof is typically the most daunting, but really amounts to nothing more than understanding the Lie algebra generated by the interaction over two tensored encoded qubits.

For the exchange interaction we have shown explicitly that nearest neighbor interactions can be used to produce such an action [2]. In particular, we find that the effect of the exchange interaction on 6 qubits becomes

$$\mathcal{L}_E^{(6)} = (\mathcal{L}_5 \otimes I_1) \oplus (\mathcal{L}_9 \otimes I_3) \oplus (\mathcal{L}_5 \otimes I_5) \oplus (\mathcal{L}_1 \otimes I_7) \tag{19}$$

When we conjoin two encoded systems, the Hilbert space of the full system contains a subspace spanned by the two encoded systems. It is the action on this conjoined subspace that we must then show can act nontrivially to produce an encoded two-qubit gate. When two of the above three-qubit codes are conjoined, we find that the resulting tensor product subspace lie entirely within the second and third irreps of the decomposition, Eq. (19). This in turn implies that non-trivial interactions between states in the encoded irreps are possible. In fact the original encoded single-qubit operations $su(2)$ are also contained within this decomposition. The most important point, however, is that on our established tensor product between encoded qubits we can indeed couple these encoded qubits in such a way as to achieve a map to the unencoded, fully universal set of gates. Thus we have established that with the exchange interaction, this encoding can efficiently (to within a factor of three in spatial resources) simulate the unencoded set of universal gates.

While the above example provides a systematic procedure to arrive at encoded universality, it is not unique. Indeed, there is a great deal of arbitrariness in how to produce a universal quantum computer using encoding. Our scheme of taking small encodings and then coupling them by conjoining is by no means the only method to obtain a universal quantum computer. An interesting alternative approach that exploits topological aspects is also under active consideration, see e.g., [28,29]. However, in dealing with real physical implementations, we expect that the use of small encodings will be very convenient. The principle of conjoining small encoded qubits therefore provides a very useful and practical route.

E. Efficiency of Encoded Gate Implementation

We now address the question of how efficiently these encoded gates can be implemented with discrete operations. Consider the encoded $su(2)$ defined by the algebra of Eq. (15). The Solovay-Kitaev theorem tells us that any encoded single-qubit operation can be implemented efficiently with some sequence of discrete gates $e^{i\mathbf{H}t_n}$, where t_n is the set of discrete times over which the exchange operations are switched on. However, it does not provide explicit values for t_n. There does exist a brute force route to such sequences of discrete gates, namely via truncation of the Trotter expansion, Eq. (2). However there is absolutely no guarantee that this expansion will be efficient.

In recent work [4] we have shown that while there does not yet appear to be an analytic route to determination of optimally efficient discrete gate sequences, it is nevertheless possible to use numerical methods to search for efficient sequences. Whether the resulting solutions are optimal or not is not known, but in the example of isotropic exchange the numerical results are striking. In particular, the numerical search in Ref. [4] showed that there exists a sequence of 19 discrete exchange gates that can achieve a nontrivial two-qubit gate between encoded blocks (the encoded CNOT), while encoded one-qubit gates require no more than 4 elementary exchange interactions. Several numerical approaches to search for efficient discrete gate sequences have been suggested [30–34]. Methods relying on local invariants of quantum states allow determination of discrete gate sequences for two-qubit encoded gates up to a one-qubit rotation [33]. This was the procedure employed in Ref. [4] where two such invariants were combined to yield a function whose minimal solution corresponds to a discrete gate sequence. Clearly there are many other approaches to the determination of optimal discrete gate sequences that could be explored, including factors such as pulse shaping and genetic algorithms.

F. A Useful Criterion for Demonstrating Non-Universality

The above discussion has focused on how one can find universal encodings deriving from a given physical interaction. While we cannot always guarantee that this is possible and in general will need to proceed on a case by case basis to find such encodings, formal conditions for the "inverse" problem can nevertheless be made. Thus, in certain cases we can indeed answer the question of whether a set of interactions *cannot* be used as a universal quantum computer, even in the presence of arbitrary encoding. We present here a useful criterion for testing this, allowing detection of Lie algebras which are not universal.

Suppose one is given a set of Hamiltonians $\mathtt{H}_n$ which can be implemented in a Hamiltonian control sequence on n subsystems. Let $\mathcal{L}_n$ denote the Lie algebra which can be generated by $\mathtt{H}_n$, and let $g(n)$ denote the number of linearly independent operators in $\mathcal{L}_n$. We call $g(n)$ the subsystems growth function.

Theorem 2 *A growth function $g(n)$ which is polynomial in n is not universal on a quantum circuit model.*

Sketch of the proof: The basic idea behind this theorem is the realization that a quantum circuit model on n subsystems has a state space which grows exponentially in n. Therefore performing unitary operators on this space is equivalent to generating elements of an exponentially growing Lie algebra.

We will illustrate the above by giving a set of Hamiltonians which is not universal for *any* encoding of the information. Consider the set of gates generated by Hamiltonians in the set

$$\mathcal{O}' = \{\sigma_z^{(i)}, \sigma_x^{(i)}\sigma_x^{(i+1)}\} \tag{20}$$

where $\sigma_\alpha^{(i)}$ is the αth Pauli matrix operating on the ith qubit tensored with identity on all other qubits. We will now show that, even with the help of encoding, this set of Hamiltonians is not universal.

Define the operation $\mathbf{M}_{jk,\alpha,\beta} = \sigma_\alpha^{(j)} \prod_{i=j+1}^{k-1} \sigma_z^{(i)} \sigma_\beta^{(k)}$ where $j < k$ and $\alpha, \beta \in \{x, y\}$. We claim that the operators in the Lie algebra generated by $\mathcal{O}'$ are all linear combinations of the form $\mathbf{M}_{jk,\alpha,\beta}$ plus the single qubit $\sigma_z^{(i)}$. Notice that this is true for $n = 2$. We will prove the result by induction. First we note that because our generators are made up of Pauli operators, we need not worry about linear combinations of operators, but only need to worry about the operators which can be generated by commutation. Let $\mathcal{L}^n$ denote the Lie algebra on n qubits generated by taking commutators in $\mathcal{O}'$. For example $\mathcal{L}_2 = \{\mathbf{M}_{12,x,x}, \mathbf{M}_{12,x,y}, \mathbf{M}_{12,y,x}, \mathbf{M}_{12,y,y}, \sigma_z^{(1)}, \sigma_z^{(2)}\}$ as claimed above. Assume that $\mathcal{L}_n = \{\mathbf{M}_{jk,\alpha,\beta}, 1 \le j < k \le n, \alpha, \beta \in \{x, y\}\} \cup \{\sigma_z^{(i)}, 1 \le i \le n\}$. First notice that taking commutators of elements of $\mathcal{L}_n$ and $\sigma_z^{(i)}$ only produces elements in $\mathcal{L}_n$: the only elements which $\sigma_z^{(i)}$ do not commute with are $\mathbf{M}_{ik,\alpha,\beta}$ and $\mathbf{M}_{ki,\alpha,\beta}$ and this commutation only serves to flip the value of α or β. Finally, note that taking the commutator between elements of $\mathcal{L}_n$ and $\sigma_x^{(i)}\sigma_x^{(i+1)}$ can only generate elements that are linear combinations of the $\mathbf{M}_{jk,\alpha,\beta}$ and $\sigma_z^{(i)}$. To see this, first note that the only nontrivial commutators are those which occur with the $\sigma_z^{(i)}$ operators, which just produce elements in $\mathcal{L}_2$. Further commutators between $\sigma_x^{(i)}\sigma_x^{(i+1)}$ and $\mathbf{M}_{jk,\alpha,\beta}$ only create $\mathbf{M}_{j'k',\alpha',beta'}$ which are one qubit larger or smaller. Thus we have proved that the Lie algebra generated by elements of $\mathcal{O}'$ is spanned by the set of linearly independent operators in $\mathcal{L}_n = \{\mathbf{M}_{jk,\alpha,\beta}, 1 \le j < k \le n, \alpha, \beta \in \{x, y\}\} \cup \{\sigma_z^{(i)}, 1 \le i \le n\}$.

Let us count the operators in $\mathcal{L}_n$. There are n $\sigma_z^{(i)}$ operators and $4\binom{n}{2}$ $\mathbf{M}_{jk,\alpha,\beta}$ operators. Thus the growth function for this Lie algebra is $g(n) = n + 4\left(\frac{n(n-1)}{2}\right) = 2n^2 - n$. This growth function is polynomial in n and thus via Theorem 2 this set of operators is not universal.

IV. ANISOTROPIC EXCHANGE

Anisotropic Heisenberg spin couplings arise whenever there is some preferred direction in space along which the coupling is stronger or weaker. This could be due to, e.g., asymmetries induced by donor atoms in solid-state arrays of atoms coupled via their nuclear spins [35,36]. The XY-interaction arises when there is no coupling in the z-direction of the spins while the coupling in x- and y-direction is equally strong:

$$(\mathbf{H}_{XY})_{ij} = \frac{J_{ij}}{2}(\sigma_x^i\sigma_x^j + \sigma_y^i\sigma_y^j) \equiv J_{ij}A_{ij}. \tag{21}$$

This situation is relevant to the proposal of solid-state quantum computation using quantum dot spins and cavity QED [37].

We will now analyze the power of the XY-interaction as a universal gate via encoding. Several recent works have addressed ways in which the XY-interaction can be made universal under the addition of either single-qubit operations [37,38], or single-qubit interactions [39]. In contrast to these proposals, our focus here is on deriving universality without any additional single-qubit interactions or operations. Our presentation here is designed not only to illustrate the general formalism and methodology of proof, but also to make the approach to find encodings that provide universality with a *single* physical interaction, transparent and applicable to other types of interactions (see also [20]). Technical details of the proofs are relegated to the appendix A.

A. Lie Algebra for Anisotropic Exchange

The set $\mathsf{H} = \{\mathsf{A}_{\mathsf{ij}} : \quad 1 \leq \mathsf{i} < \mathsf{j} \leq \mathsf{n}\}$ generates the Lie-algebra $\mathcal{L}$. To help us find the splitting into irreps Eq. (6) let us identify (by inspection) some elements in its commutant $\mathcal{L}'$:

$$S_z = \sum_{i=1}^{n} \sigma_z^i \qquad \mathbf{X} = \Pi_{i=1}^{n} \sigma_x^i \tag{22}$$

We claim that the associative algebra $\mathcal{M}$ generated by these two elements via linear combination and multiplication is identical to the commutant $\mathcal{L}'$ of $\mathcal{L}$. Note that the commutant is always closed under linear combination and multiplication and hence is an associative algebra. Together with the identity operator, $\mathcal{M}$ obviously constitutes a † closed algebra and hence satisfies the condition of Theorem 1. It splits into irreps according to Eq. (5). Assume for now that $\mathcal{L}' = \mathcal{M}$. Then the splitting of $\mathcal{L}$ and the splitting of $\mathcal{M}$ (as well as that of its commutant $\mathcal{M}'$) will be dual to each other:

$$\mathcal{L} \cong \bigoplus_{J \in \mathcal{J}} \mathcal{L}_J(n_J) \otimes I_{d_J} \quad \mathcal{M} \cong \bigoplus_{J \in \mathcal{J}} I_{n_J} \otimes M(\mathbb{C}^{d_J}) \quad \mathcal{M}' \cong \bigoplus_{J \in \mathcal{J}} M(\mathbb{C}^{n_J}) \otimes I_{d_J} \tag{23}$$

in the *same basis*. In particular the splitting of $\mathcal{M}$ would give us the *encoding* into subspaces over which H is possibly a universal set. Note that to show that H is indeed universal over these subspaces we still have to prove $su(n_J) \subseteq \mathcal{L}_J(n_J)$.

Now consider that our claim of equality between $\mathcal{L}'$ and $\mathcal{M}$ was wrong, and that actually $\mathcal{L}' \supset \mathcal{M}$ holds. This implies that $\mathcal{L} \subset \mathcal{M}'$. This would mean that the splitting of $\mathcal{M}'$ is *coarser* than the splitting of $\mathcal{L}$, since adding elements to $\mathcal{L}$ just "joins" several previously separate factors $\bigoplus \mathbb{C}^{n_J} \otimes \mathbb{C}^{d_J}$ into one bigger block $\mathbb{C}^{n_J d_J} \otimes I$. This in turn would mean that $\mathcal{L}$ in this basis would have a sub-block structure. In particular, $\mathcal{L}$ could then not generate a full $su(n_J)$ in this block because it doesn't mix certain parts of the bigger block.

Our strategy is then the following. We will first identify the splitting of $\mathcal{M}$ and its associated subspaces. Then, using an inductive proof, we shall show that over these subspaces $\mathcal{L}$ contains $su(n_J)$, thus allowing for universal computation. As a by product this will also prove our claim above that $\mathcal{L}' = \mathcal{M}$ does indeed hold.

B. The Algebra $\mathcal{M}$

Let us study the algebra generated by the two elements S_z and $\mathbf{X}$ in Eq. (22). They do not commute so we expect some irreducible representations of higher dimension. The element S_z alone is already diagonal in the standard basis, the eigenvalue of a bit-string with i ones and $n-i$ zeros is $n-2i$. The element $\mathbf{X}$ turns every zero into a one and vice versa. Let us rearrange all bit-strings into the following order

$$\mathcal{B} = \{000\ldots0, 111\ldots1, 100\ldots0, 011\ldots1, 010\ldots0, 101\ldots1, \ldots\} \tag{24}$$

i.e. listing elements with increasing number of ones and each followed by their transform under $\mathbf{X}$. In this basis S_z and $\mathbf{X}$ are block-diagonal with 2-by-2 blocks of the form $(n-2i)\sigma_z$ (for S_z) and σ_x (for $\mathbf{X}$), where $i = 0\ldots\lfloor n/2 \rfloor$ is the number of ones in the corresponding first basis bit-string of the particular block. Obviously all blocks with the same i are the same, and there are $\binom{n}{i}$ of these. By linear combination and multiplication we get the whole matrix algebra

$M(\mathbb{C}^2)$ on each block[4]. In the representation of $\mathcal{M}$ (23) we have $\mathcal{J} = \{0, 1, 2, \ldots \lfloor n/2 \rfloor\}$, $d_J = 2$ and $n_J = \binom{n}{J}$[5].

The largest encoding - assuming we can prove universality of $\mathcal{L}$ on these spaces - will be of dimension $n_J = \binom{n}{(n-1)/2}$ for odd n and $\binom{n}{n/2-1}$ for even n.[6] The dimensions of the first few encodings are given as follows:

n	1	2	3	4	5	6	7	8
max n_J	1	1	3	4	10	15	35	56

Thus, for $n = 3$ we can encode one qutrit, for $n = 4$ we can encode two qubits, for $n = 5$ one can either encode a single ten-level system, or two five-level systems, or one can use only eight of the ten available states to encode just three qubits, etc. As n gets large, the rate of encoding approaches one (i.e. $\log_2 \max n_J \to n$), i.e. the encoding approaches unit efficiency.

C. Example - One Qutrit Encoded into Three Qubits

The smallest space into which we can encode more than one state is the space of $n = 3$ qubits. This encodes a logical *qutrit*. The algebra $\mathcal{M}$ and its commutant $\mathcal{L} \subset \mathcal{M}'$ split as

$$\mathcal{M} \cong I_3 \otimes M(\mathbb{C}^2) \oplus I_1 \otimes M(\mathbb{C}^2) \qquad \mathcal{M}' \cong M(\mathbb{C}^3) \otimes I_2 \oplus M(\mathbb{C}^1) \otimes I_2. \tag{25}$$

We will show "by hand" that $\mathcal{L}$ generates all of $su(3)$ over the second product. According to the previous section we can chose the basis for our encoded qutrit as

$$|0_L\rangle = |100\rangle \quad |1_L\rangle = |010\rangle \quad |2_L\rangle = |001\rangle. \tag{26}$$

The explicit action of H on this encoded space is

$$A^L_{12} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A^L_{23} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad A^L_{13} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \tag{27}$$

where A^L_{ij} is an operator on the encoded space. Taking commutators of two of these encoded operators gives us $i\sigma_y$ on each two-dimensional subspace. Further commutators can easily be seen to generate all of $su(3)$. Hence H is a universal set on the encoded qutrit.

We can now introduce conjoining of the encoded qutrits (see Sec. III C) in order to make use of them as basic units in an encoded circuit. As outlined above, for this to be useful, we have to show that we can implement some non-trivial coupling between the qutrits such that the full $su(3 \times 3) = su(9)$ can be generated between two conjoined blocks. For example, it can easily be calculated that for the 6 qubits made by conjoining the two 3-state codes, the operator $[[A_{16}, A_{15}], A_{12}]$ produces a coupling which preserves the conjoined coding *and* non-trivially couples the two encoded qutrits. By putting together encoded qutrits (conjoining) in this manner, the XY model can therefore serve as a universal quantum computer.

Note that in general gates between encoded blocks do not constitute a problem once we have proved universality of H for higher dimensions. The 9 basis states obtained by conjoining the 2 encoded qutrits give states with exactly 2 ones (and 7 zeros). In the 6-qubit space they are all in the same subspace, namely the space generated by states with exactly 2 ones. They span a 9 dimensional subspace of this 15 dimensional space. The general proof (see below and Appendix) will show that it is possible to implement $su(15)$ on this space, and in particular the sub-algebra $su(9) \oplus su(6) \subset su(15)$, where $su(9)$ acts on the 9 states obtained by conjoining two encoded qutrits. Hence we can obtain any encoded gate between them.

---

[4]There is one exception in the case of *even* n. Over all the $\binom{n}{n/2}$ states with equal number of zeros and ones S_z acts as 0. This means that over this subspace the algebra $\mathcal{M}$ is generated by $\mathbf{X}$ only, is thus *Abelian* and splits into one-dimensional irreps in the basis where $\mathbf{X}$ is diagonal. It is easy to see that this basis is formed by states $|s\rangle + \mathbf{X}|s\rangle$ (where $\mathbf{X}$ acts as 1) and $|s\rangle - \mathbf{X}|s\rangle$ (where $\mathbf{X}$ acts as -1), where $|s\rangle$ is a bit-string. Over this subspace $\mathcal{M}$ splits into two different 1-dimensional irreps with degeneracy $\binom{n}{n/2}/2$ as $\mathcal{M} \cong I_{\binom{n}{n/2}/2} \otimes M(\mathbb{C}^1) \oplus I_{\binom{n}{n/2}/2} \otimes M(\mathbb{C}^1)$.

[5]and $n_{n/2} = \binom{n}{n/2}/2$ for even n

[6]note: $\binom{n}{n/2}/2 < \binom{n}{(n/2-1)}$ for $n > 1$

D. General Proof

We will now proceed to prove that for n qubits, where $\mathcal{M}'$ splits as[7]

$$\mathcal{M}' \cong \bigoplus_{J\in\{0,1,\ldots\lfloor n/2\rfloor\}} M(\mathbb{C}^{\binom{n}{J}}) \otimes I_2 \tag{28}$$

the Lie-algebra $\mathcal{L}$ contains $su(n_J) = su(\binom{n}{J})$ on each component. Let us call each of the associated spaces $S_n(J)$. For instance $S_3(1)$ is spanned by the three states in Eq. (26). In general we will make the canonical choice of taking our basis states to be those bit-strings which have $\#0 \geq \#1$[8]. Then for the anisotropic exchange, the label J actually counts the number of ones in the basis-bit-strings.

We will proceed by induction, as in the corresponding proof for the exchange interaction [2], to show that on each space H generates $su(n_J)$ *independently* on $S_n(n_J)$, i.e. acting trivially on the rest of the space. We have seen in the previous section that in the case of $n = 3$ H generates an *independent* $su(3)$ on $S_3(1)$ (since all three generators annihilate $S_3(0) = |000\rangle$). Let us now understand how we build bigger spaces from smaller ones when we add a new qubit. To the basis states in $S_{n-1}(J)$ we can either add 0 or 1, which gives basis states in $S_n(J)$ and $S_n(J+1)$ respectively. Turning this around this means that states in $S_n(J)$ come either from $S_{n-1}(J)$ - if a 0 was added - or from $S_{n-1}(J-1)$ - if a 1 was added[9].

For the inductive step let us assume that we can generate $su(n_J)$ *independently* on each subspace $S_{n-1}(J)$ of $(n-1)$ qubits. Within each space $S_n(J')$ of n qubits, we shall call **0** those states that come from $S_{n-1}(J')$ by adding a zero, and **1** states that come from $S_{n-1}(J'-1)$ by adding a one. We call the states derived from going back two such steps **00**, **01**, **10** and **11** states, by analogy to the TT, TB, BT and BB states used in the proof for isotropic exchange [2].

Now let us analyze how a su on $S_{n-1}(J-1)$ affects the spaces of n qubits[10]. It propagates to a su on the **0**-states of $S_n(J-1)$ and the same su on the **1**-states of $S_n(J)$. In the first step we are going to show how to eliminate the action of this su on *one* of these two spaces, keeping just the other one (*independence step*). This will give us an independent su on the **0** states of each S_n and another independent su on the **1** states. In a second step (*mixing step*) we will show how to "mix" these two su to obtain a full independent $su(n_J)$ on $S_n(J)$. These two steps can be found in the appendix A which terminates the proof of universality of the XY-interaction on encoded spaces.

E. Conjoining

As we have seen on the example of the encoded *qutrit* in Sec. IV C it is possible to obtain all encoded operations *between* the blocks, which we have to introduce to obtain a tensor-structure. Assume we cut our blocks of size n qubits to encode a qu*dit*. The XY-interaction generates any operation within a block. If we join two blocks, its basis states will give a subset of the basis-bit-strings in one of the encoded spaces over $2n$ qubits. (In fact, if we used the space $S_n(J)$ over the blocks then the bigger block will be a subspace of $S_{2n}(2J)$.) We can universally compute over this conjoined block and in particular obtain any gate in the tensor space of our encoded qudits.[11] Hence we can in principle choose any cut-off we wish and increase our coding efficiency as much as we like.

We now make some remarks on efficiency for the encoded XY. It is not clear how much overhead in time is required for actual implementation of this encoded universality with the XY-interaction. Numerical optimization studies similar to those we have made for the isotropic exchange interaction in [4] and summarized in Section III E have to be performed in order to find the minimal lengths of physical exchange gate sequences that can realize specific gates

---

[7] the last component in the even case is $M(\mathbb{C}^{\binom{n}{n/2}/2}) \otimes I_1 \oplus M(\mathbb{C}^{\binom{n}{n/2}/2}) \otimes I_1$

[8] In the even n case we will have two spaces $S_n^+(n/2)$ and $S_n^-(n/2)$ of dimension $\binom{n}{n/2}/2$, the first spanned by states of the form $|s\rangle + \mathbf{X}|s\rangle$ and the second by states of the form $|s\rangle - \mathbf{X}|s\rangle$.

[9] The only exception are the spaces $S_n^{\pm}(n/2)$ for even n. They can only be built by *adding* a 1 as the nth qubit to states in $S_{n-1}(n/2-1)$. This gives us a set of $\binom{n}{n/2}/2$ strings $|s\rangle$. We then form the linear combinations $|s\rangle \pm \mathbf{X}|s\rangle$ to obtain $S_n^{\pm}(n/2)$. We call this phenomenon *"doubling"*. Similarly, to build basis states in $S_{n+1}(n/2)$ from $S_n^{\pm}(n/2)$ we have to take all the $\binom{n}{n/2}$ bit-strings with $n/2$ ones and add a 0. We call this procedure *"rearranging"*. *"Doubling"* and *"rearranging"* will require a more careful separate analysis in the inductive proof.

[10] By this we mean a particular operation in $su(n_J)$ performed independently on this space.

[11] If we used even n and one of the spaces $S^{\pm}(n/2)$ for our encoded qudit, we have to apply a little more care with the argument. However the conjoined space will be formed by states in either of $S_{2n}(n)$ and hence there is no problem.

between encoded blocks. We note that several authors have recently shown that in a linear array, the XY-interaction is not universal when restricted to act between nearest neighbors [40–43] only. In fact this result can easily be derived using the criterion for non-universality of Sec. III F by explicitly computing all possible commutators of nearest neighbor XY-exchanges. This set turns out to be polynomial in the number of qubits. This result implies that the *geometry* of allowed interactions for our encoded universality result with the XY-interaction will be important in construction of numerical gate sequences.

V. ERROR CORRECTION AND LEAKAGE

We give here a brief discussion of the issues involved in making encoded universal computation fault tolerant. We do not pretend to give a complete treatment of this subject, but we provide some observations here that indicate that there is no fundamental obstacle to making our coded computation fully fault tolerant.

Once we have a physical embodiment of a logical qubit (e.g., the sets of three spins with the states given in Eq. (18)), and a procedure for implementing CNOT and one-qubit gates on these qubits [4], then all the fault tolerant procedures developed recently for the standard gate model for quantum computation can be applied [29]. The DFS coding would be concatenated within a stabilizer code, the appropriate transversal implementation of coded gates would be used, and appropriate error detection and correction would be implemented that limits uncontrolled error propagation. All of this carries over unchanged in the DFS setting.

However, there is one aspect of error correction for which the DFS setting is quite different, and that is in the issue of leakage. Leakage is possible because the Hilbert spaces employed in quantum computation are larger than is assumed in the ideal treatment. Ideally, the state of a qubit lives in a two-dimensional Hilbert space spanned by some states $|0_L\rangle$ and $|1_L\rangle$, and the complete Hilbert space is just a tensor product of such Hilbert spaces. But for many physical embodiments of a qubit, the actual dimension of its Hilbert space is larger, even though we may have effective ways of restricting it approximately to just two of these dimensions. For example, an atom may have two low-lying energy levels that we identify as the 0_L and 1_L states; but of course the atom will have other excited states that we might label $|2_L\rangle$, $|3_L\rangle$, etc. Unintended interactions and imperfections will surely, with some probability, bring the state of the qubit into these extra states, although we hope that the probability of this occurring is small. This straying into extra dimensions is what is referred to as leakage.

Leakage is actually a fairly likely error in the DFS schemes. For example, for the three-spin encoding of Eq. (18), The three spins naturally have an 8-dimensional Hilbert space, of which we are singling out two dimensions with quantum numbers $J = 1/2$, $J_z = +1/2$. But very simple kinds of unintended interactions, such as a magnetic field affecting just one of the three spins, will cause the qubit to "leak" into the other six dimensions (those with quantum numbers $J = 1/2$, $J_z = -1/2$, or with $J = 3/2$). So it is important to prescribe a scheme for detecting and correcting leakage to have a fully fault tolerant scheme.

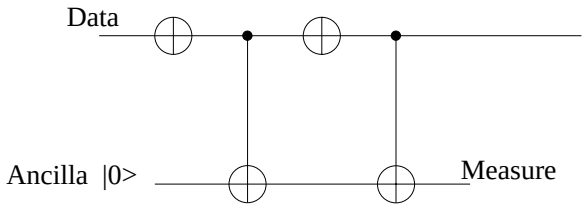

FIG. 1. Circuit suggested by Preskill for leakage detection.

Preskill [29] has indicated a simple procedure for dealing with leakage. He suggests that, from time to time, each qubit be run through the simple circuit of Fig. 1. The truth table of this circuit is to be

$$\begin{aligned} &|0_L\rangle|0_L\rangle \to |0_L\rangle|1_L\rangle \\ &|1_L\rangle|0_L\rangle \to |0_L\rangle|1_L\rangle \\ &|2_L\rangle|0_L\rangle \to |2_L\rangle|0_L\rangle \\ &\ldots \\ &|\text{leak}\rangle|0_L\rangle \to |\text{leak}\rangle|0_L\rangle. \end{aligned} \tag{29}$$

At the end of the circuit, the ancilla qubit is to be measured. If it is found to be a "1", the data qubit is in a valid state and it is unchanged by the circuit. But if it is found to be a "0", the data qubit is known to be in a "leaked"

state. In this case, the data qubit is replaced by a fresh $|0_L\rangle$ state. This is usually incorrect, but it is an ordinary bit error, and is dealt with by the error-correction machinery at a higher level.

The general strategy introduced by Preskill can still be used for the DFS-coded qubit, but the detailed procedure that he introduced will not work. From this point on we will specialize our remarks to the case of coded computation using isotropic Heisenberg exchange. The problem is that the circuit given in Fig. 1 only has the truth table given in Eq. (29) given a certain assumption about how the quantum gates in the circuit work. In particular, it is assumed, for both the NOT and the CNOT, that if one of the inputs is leaky, the gate leaves the state unchanged:

$$\begin{array}{l}\text{assumed CNOT action in Fig. 1:}\\ |0_L\rangle|0_L\rangle \rightarrow |0_L\rangle|0_L\rangle\\ |1_L\rangle|0_L\rangle \rightarrow |1_L\rangle|1_L\rangle\\ |2_L\rangle|0_L\rangle \rightarrow |2_L\rangle|0_L\rangle\\ |3_L\rangle|0_L\rangle \rightarrow |3_L\rangle|0_L\rangle\\ ....\end{array} \tag{30}$$

We find that it is not possible to implement a coded CNOT using the Heisenberg exchange that has this extended truth table, as it would require the gate to change the total spin quantum number of the state. Therefore, we must search for a different approach to leakage detection and correction.

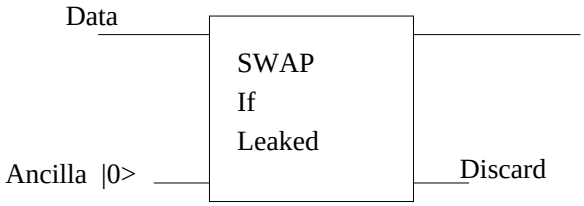

FIG. 2. Another leakage repair procedure better adapted to coded computation.

Fortunately, we find that a modified leakage-correction circuit specification is permitted by the constraints of the exchange interaction; this circuit can then be used in the same way as the Preskill leakage-correction circuit within a complete fault-tolerant procedure. In fact, the circuit action we achieve, indicated symbolically in Fig. 2, can be used in a simpler way in fault tolerant quantum computation: the leakage is corrected without the need for any measurement. What we mean by "SWAP If Leaked" (*SIL*) in Fig. 2 is the following truth table:

$$\begin{array}{l}\text{“SWAP If Leaked”, Fig. 2:}\\ |0_L\rangle|0_L\rangle \rightarrow |0_L\rangle|0_L\rangle\\ |1_L\rangle|0_L\rangle \rightarrow |1_L\rangle|0_L\rangle\\ |2_L\rangle|0_L\rangle \rightarrow |0_L\rangle|2_L\rangle\\ |3_L\rangle|0_L\rangle \rightarrow |0_L\rangle|3_L\rangle\\ ....\end{array} \tag{31}$$

We note that if the data qubit is not leaked, it is not disturbed; if it is leaked, it is always replaced by $|0_L\rangle$, permitting subsequent correction. (Of course, there is now no definite signal of when correction must be done. But if the rate of leakage is known, then error correction can be done with the appropriate frequency.)

Actually, the function of "SWAP If Leaked" is achieved with a more relaxed specification:

$$\begin{array}{l}\text{modified “swap if leaked”:}\\ |0_L\rangle|0_L\rangle \rightarrow |0_L\rangle|0_L\rangle\\ |1_L\rangle|0_L\rangle \rightarrow |1_L\rangle|0_L\rangle\\ |\text{leak}\rangle|0_L\rangle \rightarrow |\text{anything in } 0_L\text{-}1_L \text{ space}\rangle|X\rangle.\end{array} \tag{32}$$

Here $|X\rangle$ is any state vector consistent with unitarity. This generalization will be needed in our DFS analysis.

Now, we consider the implementation of *SIL* using the three-spin DFS coding discussed earlier. For this we adopt a specific notation for these states that will make the discussion easier:

$$|N, n, J, J_z\rangle. \tag{33}$$

Here N is the number of spins in the DFS (which is not really a quantum number, just a useful label), J is the total angular momentum, J_z is its projection on the z axis, and n is a "principal" quantum number that labels the different states in the DFS (e.g., n runs from 0 to 8 for the nine triplet states with $N = 6$, $J = 1$).

In this notation our two coded qubit states are

$$|0_L\rangle = |3, 0, 1/2, +1/2\rangle, \quad |1_L\rangle = |3, 1, 1/2, +1/2\rangle \tag{34}$$

We will assume that a three-spin ancilla is available that has been freshly prepared in the coded 0 state. Thus, we need to consider the operation of the *SIL* circuit on the eight basis states (the ancilla state is second):

$$\begin{array}{ll}
1. & |3, 0, 1/2, +1/2\rangle|3, 0, 1/2, +1/2\rangle \\
2. & |3, 1, 1/2, +1/2\rangle|3, 0, 1/2, +1/2\rangle \\
3. & |3, 0, 1/2, -1/2\rangle|3, 0, 1/2, +1/2\rangle \\
4. & |3, 1, 1/2, -1/2\rangle|3, 0, 1/2, +1/2\rangle \\
5. & |3, 0, 3/2, +3/2\rangle|3, 0, 1/2, +1/2\rangle \\
6. & |3, 0, 3/2, +1/2\rangle|3, 0, 1/2, +1/2\rangle \\
7. & |3, 0, 3/2, -1/2\rangle|3, 0, 1/2, +1/2\rangle \\
8. & |3, 0, 3/2, -3/2\rangle|3, 0, 1/2, +1/2\rangle.
\end{array} \tag{35}$$

3. through 8. are the "leaked" cases. To impose the constraint that the *SIL* circuit be implemented only with isotropic Heisenberg interactions, we must write these eight states in the basis of total angular momentum states for the six-spin block:

$$\begin{array}{ll}
1. & |3, 0, 1/2, +1/2\rangle|3, 0, 1/2, +1/2\rangle = |6, 0, 1, 1\rangle \\
2. & |3, 1, 1/2, +1/2\rangle|3, 0, 1/2, +1/2\rangle = |6, 1, 1, 1\rangle \\
3. & |3, 0, 1/2, -1/2\rangle|3, 0, 1/2, +1/2\rangle = 1/\sqrt{2}|6, 0, 0, 0\rangle + 1/\sqrt{2}|6, 0, 1, 0\rangle \\
4. & |3, 1, 1/2, -1/2\rangle|3, 0, 1/2, +1/2\rangle = 1/\sqrt{2}|6, 1, 0, 0\rangle + 1/\sqrt{2}|6, 1, 1, 0\rangle \\
5. & |3, 0, 3/2, +3/2\rangle|3, 0, 1/2, +1/2\rangle = |6, 0, 2, 2\rangle \\
6. & |3, 0, 3/2, +1/2\rangle|3, 0, 1/2, +1/2\rangle = a|6, 4, 1, 1\rangle + b|6, 0, 2, 1\rangle \\
7. & |3, 0, 3/2, -1/2\rangle|3, 0, 3/2, +1/2\rangle = c|6, 4, 1, 0\rangle + d|6, 0, 2, 0\rangle \\
8. & |3, 0, 3/2, -3/2\rangle|3, 0, 3/2, +1/2\rangle = e|6, 4, 1, -1\rangle + f|6, 0, 2, -1\rangle.
\end{array} \tag{36}$$

a-f are Clebsch-Gordan coefficients whose values we will not need. Recall that in this basis, the Heisenberg interactions cannot change the J quantum number, and the action of the circuit must be independent of the J_z quantum number.

The "no-leak" cases 1. and 2. require, from Eq. (31):

$$\begin{array}{l}
SIL|6, 0, 1, 1\rangle = |6, 0, 1, 1\rangle \\
SIL|6, 1, 1, 1\rangle = |6, 1, 1, 1\rangle.
\end{array} \tag{37}$$

But because of the J_z invariance of any circuit *SIL* made up only of exchanges, this implies

$$\begin{array}{l}
SIL|6, 0, 1, S_z\rangle = |6, 0, 1, S_z\rangle \\
SIL|6, 1, 1, S_z\rangle = |6, 1, 1, S_z\rangle.
\end{array} \tag{38}$$

This has implications for cases 3. and 4. We can use the decompositions

$$\begin{array}{l}
|6, 0, 1, 0\rangle = 1/\sqrt{2}|3, 0, 1/2, -1/2\rangle|3, 0, 1/2, +1/2\rangle + 1/\sqrt{2}|3, 0, 1/2, +1/2\rangle|3, 0, 1/2, -1/2\rangle \\
|6, 1, 1, 0\rangle = 1/\sqrt{2}|3, 1, 1/2, -1/2\rangle|3, 0, 1/2, +1/2\rangle + 1/\sqrt{2}|3, 1, 1/2, +1/2\rangle|3, 0, 1/2, -1/2\rangle
\end{array} \tag{39}$$

to write these out as

$$\begin{array}{l}
SIL|3, 0, 1/2, -1/2\rangle|3, 0, 1/2, +1/2\rangle = \\
(1/\sqrt{2})SIL(|6, 0, 0, 0\rangle) + 1/2|3, 0, 1/2, -1/2\rangle|3, 0, 1/2, +1/2\rangle + 1/2|3, 0, 1/2, +1/2\rangle|3, 0, 1/2, -1/2\rangle \\
SIL|3, 1, 1/2, -1/2\rangle|3, 0, 1/2, +1/2\rangle = \\
(1/\sqrt{2})SIL(|6, 1, 0, 0\rangle) + 1/2|3, 1, 1/2, -1/2\rangle|3, 0, 1/2, +1/2\rangle + 1/2|3, 1, 1/2, +1/2\rangle|3, 0, 1/2, -1/2\rangle.
\end{array} \tag{40}$$

We see that if we make the choice

$$\begin{array}{l}
SIL|6, 0, 0, 0\rangle = -|6, 0, 0, 0\rangle \\
SIL|6, 1, 0, 0\rangle = -|6, 1, 0, 0\rangle
\end{array} \tag{41}$$

(note the sign change), then using these and the relations

$$\begin{aligned} |6,0,0,0\rangle &= 1/\sqrt{2}|3,0,1/2,-1/2\rangle|3,0,1/2,+1/2\rangle - 1/\sqrt{2}|3,0,1/2,+1/2\rangle|3,0,1/2,-1/2\rangle \\ |6,1,0,0\rangle &= 1/\sqrt{2}|3,1,1/2,-1/2\rangle|3,0,1/2,+1/2\rangle - 1/\sqrt{2}|3,1,1/2,+1/2\rangle|3,0,1/2,-1/2\rangle, \end{aligned} \tag{42}$$

we can plug back in to get

$$\begin{aligned} SIL|3,0,1/2,-1/2\rangle|3,0,1/2,+1/2\rangle &= |3,0,1/2,+1/2\rangle|3,0,1/2,-1/2\rangle \\ SIL|3,1,1/2,-1/2\rangle|3,0,1/2,+1/2\rangle &= |3,1,1/2,+1/2\rangle|3,0,1/2,-1/2\rangle. \end{aligned} \tag{43}$$

Here we achieve the generalized "SWAP If Leaked" requirement of Eq. (32).

The last four cases are easier, because they are not constrained by the earlier choices. To deal with these, we introduce some new state expansions:

$$\begin{aligned} &5.\ |3,0,1/2,+1/2\rangle|3,0,3/2,+3/2\rangle = |6,2,2,2\rangle \\ &6.\ |3,0,1/2,+1/2\rangle|3,0,3/2,+1/2\rangle = a|6,7,1,1\rangle + b|6,2,2,1\rangle \\ &7.\ |3,0,1/2,+1/2\rangle|3,0,3/2,-1/2\rangle = c|6,7,1,0\rangle + d|6,2,2,0\rangle \\ &8.\ |3,0,1/2,+1/2\rangle|3,0,3/2,-3/2\rangle = e|6,7,1,-1\rangle + f|6,2,2,-1\rangle. \end{aligned} \tag{44}$$

Note that the Clebsch-Gordan coefficients *a*-*f* are the same as above. Then, we can fix the action of *SIL* to be

$$\begin{aligned} SIL|6,0,2,J_z\rangle &= |6,2,2,J_z\rangle \\ SIL|6,4,1,J_z\rangle &= |6,7,1,J_z\rangle, \end{aligned} \tag{45}$$

which will give us

$$\begin{aligned} SIL|3,0,3/2,+3/2\rangle|3,0,1/2,+1/2\rangle &= |3,0,1/2,+1/2\rangle|3,0,3/2,+3/2\rangle \\ SIL|3,0,3/2,+1/2\rangle|3,0,1/2,+1/2\rangle &= |3,0,1/2,+1/2\rangle|3,0,3/2,+1/2\rangle \\ SIL|3,0,3/2,-1/2\rangle|3,0,1/2,+1/2\rangle &= |3,0,1/2,+1/2\rangle|3,0,3/2,-1/2\rangle \\ SIL|3,0,3/2,-3/2\rangle|3,0,1/2,+1/2\rangle &= |3,0,1/2,+1/2\rangle|3,0,3/2,-3/2\rangle, \end{aligned} \tag{46}$$

as desired. This completes the analysis. We see that the specifications of Eqs. (38,41,45) for the desired circuit are for a unitary transformation that is obtainable by only Heisenberg interactions, by the coded universality theorem of [2]. We have not determined an explicit circuit for *SIL*; it might be interesting to use the techniques of [4] to find one. Note that the specifications Eqs. (38,41,45) for *SIL* only constrain its action on 16 of the 64 dimensions of the Hilbert space; its action on the remaining 48 dimensions is completely arbitrary, except for the constraints of unitarity.

VI. OUTLOOK AND OPEN QUESTIONS

It is unclear at this point which general kinds of physical interactions are amenable to "encoded universality". As we have seen, both the exchange and the XY-interaction give rise to a Lie-algebra that splits into degenerate irreducible representations. The degeneracy of this decomposition is the key to finding a suitable encoding. In physics, mathematical degeneracies are often linked to symmetries of a system, and indeed both the exchange and the XY-interaction possess a significant amount of symmetry. (The isotropic exchange is invariant under permutation of the Pauli matrices, and the XY-interaction is invariant under permuting σ_x and σ_y.) On the other hand, we know from previous work of Lloyd and others [15] that each *generic* (and in particular non-symmetric) two-qubit interaction is universal *without* encoding if we allow bits to be flipped (i.e., the exchange gate). Note that both the exchange gate and the XY-interaction are invariant under the permutation of the specific two qubits on which they act. It would be interesting to know what the "encoded power" of other interactions not supplemented by an exchange interaction is, and also how this power is related to their symmetry properties.

Another interesting question is related to the topology and arrangement of the physical qubits of the system. In many cases only nearest-neighbor interactions can be enacted. In our formalism we have allowed for application of the gate between arbitrary pairs of qubits. Thus, it would be important to know whether a restricted set of gates, for instance, only between qubits i and $i+1$, has the same universality power. Clearly, in the case of the exchange interaction, we can limit ourselves to only nearest neighbor interactions, because we can decompose all permutations into such pairs. Thus exchange can be used to flip qubits at our convenience. However, as indicated above (Section IV E), this is not true for the case of the XY-interaction. Whether for example certain two-dimensional layouts might be sufficient is still unclear.

A third open question is concerned with the overhead in time, i.e., the number of gates involving physical interactions switched on for discrete time intervals that are needed to realize an "encoded" standard gate. As discussed in Section III E, there is currently no analytic route to finding the optimal gate sequences and we cannot even estimate the length of this in the general case. Establishing optimal discrete sequences is currently dependent on numerical searches in many dimensions. It would be highly desirable to develop systematic bounds on the length of sequences, even if an analytical route to optimization remains elusive.

Acknowledgements: JK, DB, and KBW's effort is sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-01-2-0524. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Laboratory, or the US Government. DPD's effort was supported by the National Security Agency (NSA) and the Advanced Research and Development Activity (ARDA) under Army Research Office (ARO) contract number DAAG55-98-C-0041.

APPENDIX A: DETAILS OF THE UNIVERSALITY PROOF FOR THE XY INTERACTION

Claim: Independence – Using operations in the independent su on $S_{n-1}(J-1)$ and $A_{n-1,n}$ we can generate an independent su on the **1**-states of $S_n(J)$.

Proof: Use two **0** states in $S_{n-1}(J-1)$ and a σ_z on them. This propagates to a σ_z on the **01**-states of $S_n(J)$ and to a σ_z on the **00**-states of $S_n(J-1)$. Commute this σ_z with $A_{n-1,n}$. Since $A_{n-1,n}$ annihilates **00**-states the net action on $S_n(J-1)$ will be zero, whereas the net action on $S_n(J)$ will be $i\sigma_y$ between one of the **01**-states and another **10** state. Commute again with $A_{n-1,n}$ to obtain an *independent* σ_z on the original two **01**-states in $S_n(J)$. This new σ_z, now entirely acting only on $S_n(J)$, can be used via commutation with operations in the su to generate the now *independent* full su on the **1**-states of $S_n(J)$ with the same arguments as in [2] (using the Mixing Lemma of App. C there).

When does this method not work? Only if there are less than 2 **0**-states in $S_{n-1}(J-1)$. If $n-1 \geq 3$ this is never the case except for $S_{n-1}(0)$, which just contains the $|00\ldots0\rangle$ state. In this case $S_n(1)$ just contains one **1**-state and $su(1) = 0$ is trivially implemented on it[12].

Of course given independent su on the **1**-states of S_n we can immediately also get an independent su on the **0**-states by subtracting the former from the coupled action of a su on a S_{n-1}.

Claim: Mixing – Given an independent su on each of the **0**-states resp. **1**-states of S_n closing the Lie-algebra with $A_{n-1,n}$ gives the full su on S_n (independent).

Proof: To get a $i\sigma_y$ between a **0**-state and a **1** state take a σ_z between two **10** states (or between a **00** and a **10**-state) and commute it with $A_{n-1,n}$. To get a σ_z between these two states, commute with $A_{n-1,n}$ again. This gives a $su(2)$ connecting **0**- and **1**-states. The result follows from the Enlarging Lemma in [2] App. C. This proof works for all sizes of S_n.

---

[12] Let us treat the case of even n here: *"Doubling"* – If we can implement su on $S_{n-1}(n/2-1)$ by construction this gives us coupled simultaneous su's on the two spaces $S_n^{\pm}(n/2)$ together with the same su on the **0**-states of $S_n(n/2-1)$. The action on these latter states can be eliminated as before and we are left with these two coupled full su's on $S_n^{\pm}(n/2)$. We will not attempt to eliminate the action of one of them, for practical purposes, if we need to use these spaces, we will only be able to use one of them. *"Rearranging"* – The problematic space is $S_{n+1}(n/2)$, more precisely its **0**-states. A priory we only obtain two coupled smaller su's on this **0**-space, and not the full su. However since rearranging provides a change of basis we do not have a block-structure and it is easy to see that commuting with $A_{n-1,n}$ mixes the two su's to give the full su on the **0**-states of $S_{n+1}(n/2)$.

[1] D. Bacon, J. Kempe, D.A. Lidar, and K. Whaley, Phys. Rev. Lett. **85**, 1758 (2000).
[2] J. Kempe, D. Bacon, D.A. Lidar, and K. Whaley, Phys. Rev. A **63**, 042307 (2001).
[3] D. Bacon *et al.*, in *Proceedings of the International Conference on Experimental Implementation of Quantum Computation (IQC'01)* (Rinton Press, Australia, 2001), LANL preprint quant-ph/0102140.
[4] D. P. DiVincenzo *et al.*, Nature **408**, 339 (2000).
[5] A. Dewdney, *The Turing Omnibus: 61 Excursions in Computer Science* (Computer Science Press, Rockville, MD, 1989).
[6] E. Bernstein and U. Vazirani, SIAM Journal on Computing **26**, 1411 (1997).
[7] D. Deutsch, Proc. Roy. Soc. London Ser. A **400**, 96 (1985).
[8] D. Deutsch, Proc. Roy. Soc. London Ser. A **425**, 73 (1989).
[9] R. Feynman, Optics News, February **11**, (1985).
[10] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2000).
[11] D. P. DiVincenzo, Phys. Rev. A **51**, 1015 (1995).
[12] T. Toffoli, in *Automata, Languages and Programming*, edited by W. de Bakker and J. van Leeuwen (Springer, New York, 1980), p. 632, technical Memo MIT/LCS/TM-151, MIT Lab. for Comp. Sci. (unpublished).
[13] D. Barenco *et al.*, Phys. Rev. A **52**, 3457 (1995).
[14] D. Deutsch, A. Barenco, and A. Ekert, Proc. Roy. Soc. London Ser. A **449**, 669 (1995).
[15] S. Lloyd, Phys. Rev. Lett. **75**, 346 (1995).
[16] P. Shor, in *Proceedings of the 37th Symposium on Foundations of Computing* (IEEE Computer Society Press, Los Alamitos, CA, 1996), p. 56.
[17] P. Zanardi and M. Rasetti, Phys. Rev. Lett. **79**, 3306 (1997).
[18] D.A. Lidar, I. Chuang, and K. Whaley, Phys. Rev. Lett. **81**, 2594 (1998).
[19] P. O. Boykin *et al.*, in *Proceedings of 40th Annual Symposium on the Foundations of Computer Science (FOCS)* (IEEE Press, Los Alamitos, CA, 1999), p. 486.
[20] J. Kempe, Universal Noiseless Computation: Mathematical Theory and Applications, 2001, Ph.D. thesis, University of California, Berkeley.
[21] D. Bacon, 2001, Decoherence, Control, and Symmetry in Quantum Computers, Ph.D. thesis, University of California, Berkeley.
[22] L. Viola, E. Knill, and S. Lloyd, Phys. Rev. Lett. **85**, 3520 (2000).
[23] E. Knill, R. Laflamme, and L. Viola, Phys. Rev. Lett. **84**, 2525 (2000).
[24] N. Landsman, Lecture Notes on C*-algebras, Hilbert C*-modules and Quantum Mechanics, LANL preprint math-ph/9807030.
[25] P. Zanardi, Phys. Rev. A **63**, 012301 (2001).
[26] J. Cornwell, *Group theory in physics* (Academic Press, New York, 1984), Vol. I-II.
[27] R. Raussendorf and H. Briegel, Phys. Rev. Lett. **86**, 5188 (2001).
[28] A. Kitaev, Fault-tolerant quantum computation by anyons, 1997, LANL preprint quant-ph/9707021.
[29] J. Preskill, in *Introduction to quantum computation*, edited by T. S. H.K. Lo, S. Popescu (World Scientific, New Jersey, 1997), p. 213.
[30] E. Rains, 1997, LANL preprint quant-ph/9704042.
[31] M. Grassl, M. Roetteler, and T. Beth, Phys. Rev. A **58**, 1833 (1998).
[32] G. Burkard, D. Loss, D. P. DiVincenzo, and J. A. Smolin, Phys. Rev. B **60**, 11404 (1999).
[33] Y. Makhlin, Nonlocal properties of two-qubits Gates and Mixed States and Optimization of Quantum Computations, 2000, LANL preprint quant-ph/0002045.
[34] G.D. Sanders, K.W. Kim, and W.C. Holton, Phys. Rev. A **59**, 1098 (1999).
[35] B. Kane, Nature **393**, 133 (1998).
[36] B. Kane, Silicon-based Quantum Computation, 2000, LANL preprint quant-ph/0003031.
[37] A. Imamoglu *et al.*, Phys. Rev. Lett. **83**, 4204 (1999).
[38] L.-A. Wu and D.A. Lidar, 2001, LANL preprint quant-ph/0103039.
[39] D.A. Lidar and L.-A. Wu, 2001, Lanl preprint quant-ph/0109021.
[40] L.G. Valiant, in *Proceedings of the 33rd ACM Symposium on the Theory of Computation (STOC)*, El Pso, Texas, 2001, ACM Press.
[41] B. Terhal and D. P. DiVincenzo, 2001, LANL preprint quant-ph/0108010.
[42] E. Knill, 2001, Lanl preprint quant-ph/0108033.
[43] L.-A. Wu and D.A. Lidar, 2001, LANL preprint quant-ph/0109078